(1)

WRDC-TR-90-8007
Volume V
Part 39

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 39 - CDM Impact Analysis Development Specification

M. Apicella, S. Singh

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

David Judson

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:

Bruce Rasmussen

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release;
Distribution is Unlimited,

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
DS 620341420

5. MONITORING ORGANIZATION REPORT NUMBER(S)
WRDC-TR-90-8007 Vol. V, Part 39

6a. NAME OF PERFORMING ORGANIZATION
Control Data Corporation;
Integration Technology Services

6b. OFFICE SYMBOL *(if applicable)*

7a. NAME OF MONITORING ORGANIZATION
WRDC/MTI

6c. ADDRESS *(City,State, and ZIP Code)*
2970 Presidential Drive
Fairborn, OH 45324-6209

7b. ADDRESS *(City, State, and ZIP Code)*
WPAFB, OH 45433-6533

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Wright Research and Development Center,
Air Force Systems Command, USAF

8b. OFFICE SYMBOL *(if applicable)*
WRDC/MTI

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM.
F33600-87-C-0464

8c. ADDRESS *(City, State, and ZIP Code)*
Wright-Patterson AFB, Ohio 45433-6533

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
78011F	595600	F95600	20950607

11. TITLE *(Include Security Classification)*
See block 19 fication

12. PERSONAL AUTHOR(S)
Control Data Corporation: Apicella, M. L., Singh, S.

13a. TYPE OF REPORT
Final Report

13b. TIME COVERED
4/1/87-12/31/90

14. DATE OF REPORT *(Yr.,Mo.,Day)*
1990 September 30

15. PAGE COUNT
38

16. SUPPLEMENTARY NOTATION
WRDC/MTI Project Priority 6203

17. COSATI CODES

FIELD	GROUP	SUB GR.
1308	0905	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify block no.)*

19. ABSTRACT *(Continue on reverse if necessary and identify block number)*

This Development Specification (DS) establishes the performance, development, test, and qualification requirements of the CDM Impact Analysis computer program.

BLOCK 11:

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol V - Common Data Model Subsystem

Part 39 - CDM Impact Analysis Development Specification

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
David L. Judson

22b. TELEPHONE NO. *(Include Area Code)*
(513) 255-7371

22c. OFFICE SYMBOL
WRDC MTI

EDITION OF 1 JAN 73 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface,and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION 1

SCOPE

1.1 Identification

This specification establishes the performance, development, test and qualification requirements of a computer program identified as the CDM Impact Analysis. The CDM Impact Analysis is one configuration item of the Integrated Information Support System (IISS).

Please refer to the Software Availability Bulletin, Volume III, Part 16, CI# SAB620326000, for current IISS software and documentation availability.

1.2 Functional Summary

This Computer Program Configuration Item (CPCI) is used to identify and report which software modules are affected by a change to the CDM and to identify and report affected external schemas used by these software modules.

SECTION 2

DOCUMENTS

2.1 Reference Documents

[1] Systran, ICAM Documentation Standards, 23 December. 1981, IDS150120000A.

[2] D. Appleton Company, CDM Administrator's Manual, UM 620341000, 29 April 1985.

[3] D. Appleton Company, CDM1, An IDEF1 Model of theCommon Data Model, CCS620341000, 15 May 1985.

[4] Control Data Corporation, Neutral Data Definition Language User's Guide, UM620341100, 31 May 1988.

[5] C. J. Date, An Introduction to Database Systems, 1977, Addison-Wesley Publishing Company, Inc.

[6] Control Data Corporation, NDDL Development Specification, DS620341100, 31 May 1988.

[7] Control Data Corporation/D. Appleton Company, Conceptual Schema of CDM-1 for Cross References, 4 December 1985.

2.2 Terms and Abbreviations

Application Process: (AP), a cohesive unit of software that can be initiated as a unit to perform some function or functions.

Common Data Model: (CDM), IISS subsystem that describes common data of an enterprise and includes conceptual, external and internal schemas and schema transformation operators.

Common Data Model Administrator: (CDMA), the person or group of persons responsible for creating and maintaining an enterprises Common Data Model. The CDMA manages the common data rather than managing applications that access data.

Common Data Model Processor: (CDMP), a component of the Common Data Model subsystem which is the distributed database manager of the IISS.

Conceptual Schema: (CS), the standard definition used for all data in the CDM. It is based on IDEF1 information modelling.

External Schema: (ES), an application's view of the CDM's conceptual schema.

Integrated Information Support System: (IISS), a computing environment used to investigate, demonstrate, test the concepts and produce application for information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Internal Schema: (IS), the definition of the internal model, the storage structure definition, which specifies how the physical data are stored and how they can be accessed. It is represented in terms of the physical database components, including record types and inter-record relationships.

Neutral Data Definition Language: (NDDL), a language used to manipulate and populate information in the Common Data Model (CDM) or IISS System Database.

Neutral Data Manipulation Language: (NDML), a language developed by the IISS project to provide uniform access to common data, regardless of database manager or distribution criteria. It provides distributed retrieval single node update, and non-guaranteed distributed update.

Presentation Schema: (PS), the totality of the form fields in an application which are targets of data derivative from the common data.

SECTION 3

REQUIREMENTS

3.1 Computer Program Definition

The CDM Impact Analysis will report software modules using external schemas affected by CDM changes. The generated reports are to be used to keep application programs in a consistent state. The report will contain: the name of the software module affected, the type of NDDL command causing the change and the ES affected by the command (view name and data items). Since some NDDL commands indirectly affect external schemas the report includes the sequence of the objects affected, from the objects directly affected by the NDDL command to the external schema used by a software module. This will provide the CDM administrator with enough information to either modify software modules to work with the new CDM model or revise the NDDL changes themselves.

3.1.1 System Capacities

The system capacities of the CDM Impact Analysis have not been determined.

3.1.2 Interface Requirements

The sequence of events to produce the report is as follows:

1. NDDL processor front end and check NDDL command for software module impact yields: schema objects directly affected by command.

2. Trace schema object to external schema yields: list of external schema objects affected directly or indirectly by command.

3. Compare ES objects affected with ES objects used yields are part of software modules using affected objects.

Additional input to step two consists of the CDM data base being changed. Additional input to step three consists of a cross reference table (software module/ES) which is constructed during precompilation of a software module.

3.1.2.1 Interface Block Diagram

```
+----------------------------+
|        NDDL Source         |
+-------------+--------------+
              |
              V
******************************
* NDDL Processor front end   *
*************+****************
* determine objects affected *
*    by NDDL command         *
*************+****************
              |
              V
+----------------------------+
|   Schema objects directly  |
|     affected by command,   |
|    (model of data base)    |
+-------------+--------------+
              |
              V
******************************
*   trace schema object to   *
*      external schema       *
*************+****************
              |
              V
+----------------------------+
|   External schema objects  |
|      affected by command,  |
|   trace of command object  |
|     to external schema,    |
|       (cross reference,    |
|     application to ES)     |
+-------------+--------------+
              |
              V
*****************************
* Populate CDM tables and   *
*     Generate Reports      *
*************+***************
              |
              V
```

Figure 3-1(a). Interface Block Diagram

```
                |
                V

+-----------------------------+
|   Report of applications    |
|   using affected external   |
|       schema objects        |
+-----------------------------+

   ***********     +---------+
   * process *     | object  |
   ***********     +---------+
```

Figure 3-1(b). Interface Block Diagram (Concluded)

3.1.2.2 Detailed Interface Definition

Each step of the CDM Impact Analysis is entirely completed before the next step starts. The output of one step is the input for the next. The output of one step is ignored by following steps unless it is passed explicitly to the next step.

NDDL commands are input to the NDDL processor front end. These commands are in the form of either a text file or are entered interactively to the CDM Impact Analysis. As each command is parsed it is checked to see if it has impacts. If it does an entry is made in the object's list.

The input to the second step consists of lists of the directly affected schema objects. These objects are Data_Base, Record_Type, Data_Field, Record_Set, Attribute_Use_Cl, Relation_Class, Data_Item, and User_View. This step also populates the tables Commands, Object_Com_Trace and Application_Impact. These tables are documented in section 3.5.1.

The input to the third step consists of these same lists of schema objects with the change impacts propagated to the external schema. Additional input to the third step consists of the database tables Commands, Object_Com_Trace, and Application_Impact.

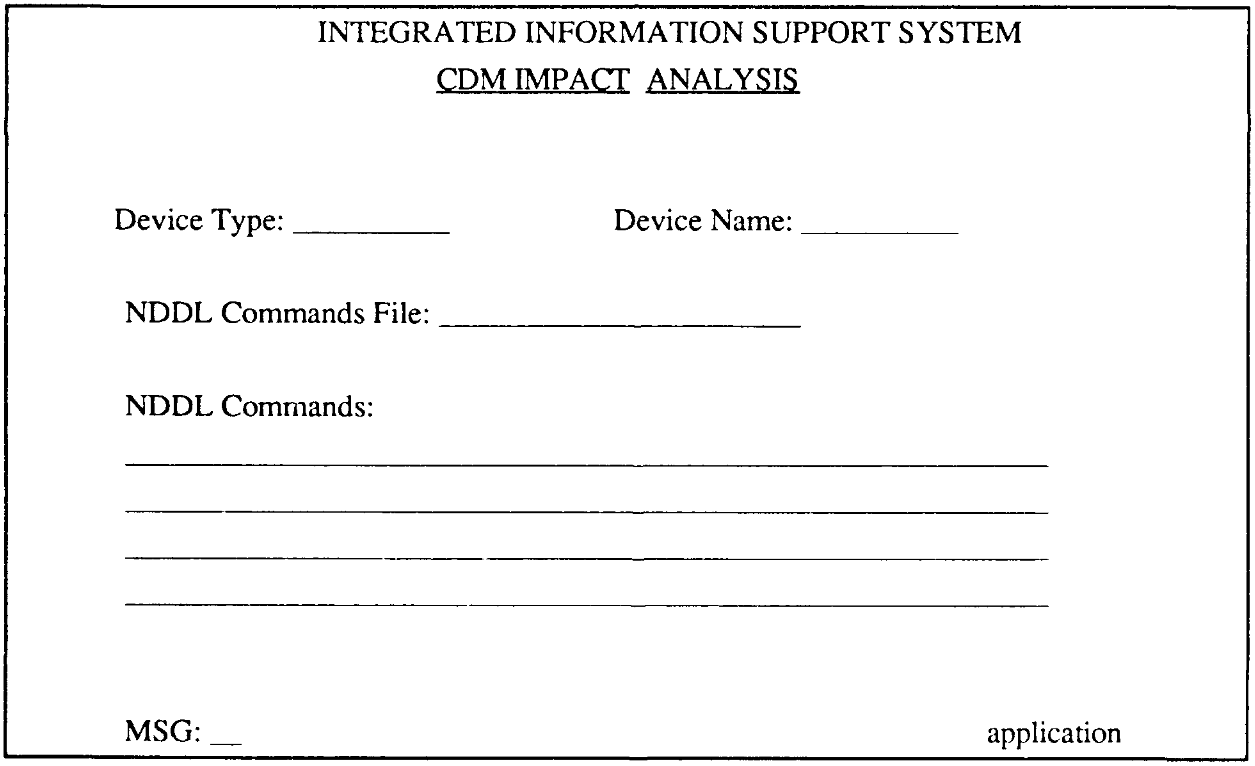
3.2 Detailed Function Requirements

Software for step one. The front end of the NDDL processor is required. This is the part that parses the commands and constructs parse tree data structures. This software is sufficiently isolated from the rest of the NDDL processor that it can be used without modification. The second purpose of this module is to examine each command to determine which of the objects it affects would necessitate reprecompilation of a software module. This includes objects which are affected directly and indirectly.

Software for step two. The purpose of this step is to determine all the external schema objects affected by a command. It takes as input all the objects directly affected by an NDDL command and, using the model of the data base as a guide, traces connections to external schema objects. It also populates the database table listed in section 3.5.1. This procedure is based on the IDEF1 model of the CDM and the NDDL DS.

Software for step three. This step takes as input the ES objects affected by an NDDL command produced by step two and the application/ES cross reference produced at precompilation time. The inputs are presented as tables so this module may be specified by the application generator. The output is the report listing applications which must be reprecompiled, what ES objects are affected and the type of NDDL command which caused the change. Optionally the internal and conceptual schema objects affected by this command can be listed in a separate report.

The input to the CDM Impact Analysis consists of NDDL commands. These may come from a file or may be input directly to the Analysis. The input fields "Device Type" and "Device Name" are similar to those on the IISS function screen and allow the user to indicate the type of output device and the name of the device the reports will be set to. Blank fields indicate the user's terminal. Figure 3-2 illustrates the form which will be presented to the user for executing the CDM Impact Analysis.



INTEGRATED INFORMATION SUPPORT SYSTEM

CDM IMPACT ANALYSIS

Device Type: __________ Device Name: __________

NDDL Commands File: ______________________

NDDL Commands:

____

____

____

____

MSG: __ application

Figure 3-2 CDM Impact Analysis User Interface Form

3.2.1 NDDL Processor Front End

The parser is that used by the NDDL command processor. When a command is recognized, a procedure is called which processes that command. This step also determines the effects of an NDDL command. The impact analysis does not execute the commands or perform the error checks of the NDDL processor. In addition to the direct effects of a command there are also ripple effects. The direct effects of a command are the objects named directly in the command. The ripple effects are indirect effects caused by the command. For most commands these are detected by step two software. This step is concerned with the case of inherited attributes. The following entities constitute directly affected objects:

Data_Base
Record_Type
Data_Field
Record_Set
Attribute_Use_Cl
Relation_Class
Data_Item
User_View

Each of these objects is associated with a list of specific instances which are impacted. These lists are stored in memory. The data structure given in Figure 3-3 illustrates an element of these lists. The arrow (-->) indicates a pointer to the next element of the list. The obj_type and obj_name are one of those listed in table Obj_Com_trace in section 3.5.1. The data field "id_no" is a value which uniquely identifies the instance of the object for retrieval from the CDM. The processing required to obtain this value for each NDDL command is given in appendices A and B. The com_no is the number of the report_message as described in section 3.5.1.

next_object
object_name
id_no
com_no

Figure 3-3 Data Structure of Impacted Schema Object

The impacts to the conceptual schema are handled by special procedures. This is necessary to trace the impacts due to inherited attributes. There are four types of entities which could cause an impact: attribute use class, key class, relation class and entity class. The following procedures handle the effects of changes to these four types of entities. There is a procedure for each type and the narrative describes what the processing will be to determine the impacts due to inherited attributes. This is in terms of what the input parameters are and what tables for step three are to be populated. In the following a percent sign (%) preceding a name indicates the procedure's parameter value.

1. AUC (tag_no, ac_no, include_root) - For processing, select tag_no from ATTRIBUTE_USE_CL using %ac_no. Next, insert tag_no into attribute use class object where tag_no is strictly an inherited attribute of %tag_no. if %include_root is true then also insert %tag_no.

2. KC (kc_no) - For processing, select ac_no, tag_no from ATTRIBUTE_USE_CL, KEY_CLASS_MEMBER using %kc_no. For each tag_no, call AUC with tag_no and include_root = FALSE to find other affected attribute use class entities.

3. RC (rc_no, kc_no) - For processing, insert rc_no into relation class object. Next, select ac_no, tag_no from ATTRIBUTE_USE_CL, INHERITED_ATT_USE using %rc_no and %kc_no. For each tag_no, call AUC with tag_no and include_root = TRUE.

4a. EC (ec_no) - For processing, select rc_no from RELATION_CLASS using %ec_no equal to dependent entity or independent entity. Next, for each rc_no, insert c_no into relation class object. Next, select ac_no, ag_no from ATTRIBUTE_USE_CL using %ec_no. Next, for each tag_no, call AUC with tag_no and include_root = TRUE to find other affected attribute use class entities.

4b. ECAUC (ec_no) - This differs from EC in that only the attributes of an entity are impacted. For processing, select tag_no from ATTRIBUTE_USE_CL using %ec_no. Next, for each tag_no insert tag_no into attribute use class object.

The NDDL commands which could be input and their impacts are documented in Appendix A.

3.2.2 Trace Impact of NDDL Commands

This is a list of the common sequences which trace the effect of a change to an object in a schema to the data items and user views of the related external schema. In general they always follow the same path. Each object on the path is represented by a data structure (see Figure 3-3). A procedure processes each type of object to propagate effects of a change from one level in Figure 3-4 to the next. For each object/procedure the following are listed: object name, and the keys/parameters. The symbol "-->" means continue with the named sequence.

```
1.   Data_Base (DB_ID)
   --> Record_Type (RT_NO)
   --> Record_Set (SET_NO)

2.   Record_Type (RT_NO)
     --> Data_Field (DF_NO)

3.   Data_Field (DF_NO)
     Project_Data_Field (DB_ID, DF_ID, RT_ID)
     --> Attribute_Use_Class (TAG_NO)

4.   Record_Set (SET_NO)
     AUC_ST_Mapping (SET_ID, DB_ID)
     --> Attribute_Use_Class (TAG_NO)
     RC_Based_Rec_Set (SET_ID, DB_ID)
     --> Relation_Class (RC_NO)

5.   Attribute_Use_Class (TAG_NO)
     Project_Data_Item (TAG_NO)
     --> Data_Item (DI_NO)

6.   Relation_Class (RC_NO)
     View_EC_XREF (RC_NO, VIEW_NO)
     --> User_View (VIEW_NO)

7.   Data_Item (DI_NO)
     select software module using %DI_NO and
     insert <software module, DI_NO> into
     Application_Impact

8.   User_View (VIEW_NO)
     select software module using %VIEW_NO and
     insert <software module, VIEW_NO> into
     Application_Impact
```

Note: There are two record set object effects: one maps to Attribute Use Class and the other to Relation Class.

Figure 3-4 Trace Impacts to External Schema

This step also populates the Object_Com_Trace table. This table is used in the report output to show what objects are affected by which commands. This table contains the command, object name, object type, and a schema sequence number for ordering the rows.

3.2.3 Report Output Specification

The report output consists of the name of the software module impacted, the ES objects used by it, user view, and data item, and the command which caused the impact. A second report lists the command, the schema object type and name affected by it, where a schema object is one of those listed in 3.2.1.

The entities used from the CDM cross reference include Software_Module, NDML_Module, View_Usage, and Data_Item_Usage.

For the reports listed below the following NDDL command has been issued:

DROP VIEW ENGRG_PART;

3.2.3.1 Impacted Software Modules Report

The format for this report is:

```
MM/DD/YY              Impact of NDDL Commands                  Page nnn


Module          User View/Data Item                    Commands
A11             ENGRG_PART.ENG_PART_ID                 DROP VIEW ENGRG_PART
                ENGRG_PART.LATEST_APPROVED_REV         DROP VIEW ENGRG_PART
                ENGRG_PART.ENG_PART_DESC               DROP VIEW ENGRG_PART
                         •
                         •
                         •
```

Figure 3-5 Report Format of Impact of NDDL Commands

This report gets its input from the Commands and Application_Impact tables. The report is sorted by software module, user_view/data_item and command. The software module is blank if it is the same as the previous line.

3.2.3.2 Impacted Schema Objects Report

The format for this report is:

Command Object_Type Object_Name

```
MM/DD/YY              Trace of NDDL Commands                   Page nnn


Objtype         Object Name                            Commands
USER VIEW       ENGRG_PART                             DROP VIEW ENGRG_PART
DATA ITEM       ENGRG_PART.ENG_PART_ID                 DROP VIEW ENGRG_PART
DATA ITEM       ENGRG_PART.LATEST_APPROVED             DROP VIEW ENGRG_PART
DATA ITEM       ENGRG_PART.ENG_PART_DESC               DROP VIEW ENGRG_PART
                         •
                         •
                         •
```

Figure 3-6 Report Format of Trace of Impacts of NDDL Commands

This report gets its input from the Commands and Object_Com_Trace tables. The report is sorted by Sequence_no, Command and Object_Name. The Command is blank if it is the same as the previous line. The object types and name formats are listed in section 3.5.1 under the description of table: Object_Com_Trace.

3.3 Performance Requirements

3.3.1 Programming Method

The front end of the NDDL command processor will be used to parse the NDDL command which may be input from the terminal or from a file and create parse tree data structures. The module which determines the external schemas affected by the command will leave its results in two CDM tables. The Report Writer will be used to read the CDM tables and present the formatted report which can be sent to a hardcopy or to a file.

3.3.2 Modification Considerations

This CPCI is dependent on the internal structure of the CDM as described in the documents [3] and [6]. The internal structure dependencies are listed for each NDDL command in the form of entities and tables. When modifications are made to the structure of the CDM, this document should be examined to determine if the impact analysis requires modification.

3.3.3 Expandability

Expansion will be required as new commands are added to the NDDL. Each command must be studied to determine its effects on application programs. A new command procedure must be written to populate the data structure given in figure 3-3. It is expected that this can be done without modification to the Report Writer section.

3.4 Human Performance

The performance requirements of the CDM Impact Analysis have not been determined.

3.5 Data Base Requirements

3.5.1 Internal Tables and Parameters

The tables used in the CDM are described in [6]. The report also makes use of the cross reference application/ES, documented in [7]. The CDM Impact Analysis populates the following tables. These tables must be defined to the CDM by the NDDL command processor. When the CDM Impact Analysis is started the rows of each of the following tables are deleted.

Table	Column
Commands	com_no
	rpt_message
	user_id

This table contains a list of NDDL commands in the form of the report messages as described in appendices A and B. The com_no is a number used to identify the command.

Table	Column
Objcomtrace	com_no
	obj_type
	obj_name
	sequence_no
	user_id

This table contains the trace of the objects affected by an NDDL command. The com_no identifies the command in the Commands table causing the impact. The sequence_no is an aid for ordering the objects from internal schema through the external schema. The obj_type and obj_name identify the object being affected. The form and values of sequence_no, obj_type, and obj_name are as follows:

Sequence	Obj Type	Obj Name
1	DATABASE	db_id
2	RECORD TYPE	db_id.rt_id
3	DATA FIELD	db_id.rt_id.df_id
3	RECORD SET	db_id.set_id
4	DF/AUC MAP	db_id.rt_id.df_id/ec_name.tag_name
4	SET/AUC MAP	db_id.set_id/ec_name.tag_name
4	SET/RC MAP	db_id.set_id/ ind_ec_name rc_name dep_ec_name
5	ENTITY CLASS	ec_name
5	ATTRIBUTE CL	ac_name
5	AUC	ec_name.tag_name
5	RELATION CL	rc_name
5	KEY CLASS	ec_name.kc_name
6	AUC/DI MAP	ec_name.tag_name/view_id.di_id
6	RC/UV MAP	ind_ec_name rc_name dep_ec_name/view_id
7	DATA ITEM	view_id.di_id
7	USER VIEW	view_id

Table	Column
Appimp	software_mod view_di_id com_no user_id impact_luw

This table contains the information for the impact report. The software_mod is the name of the procedure which is impacted. The view_di_id is the external schema used by software_mod. The com_no identifies the command impacting the software_mod. The impact_luw identifies the logical unit of work containing the software_mod.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definition

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test will consist of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests will be performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs".

Testing requires a CDM containing databases and precompiled application programs which form a representative sample from the Integrated Sheet Metal Center environment. An instance of each NDDL command which causes and does not cause impacts to application programs will be submitted to the CDM Impact Analysis. The evaluation will consist of the comparison of the expected results with those of the CDM Impact Analysis. The tests will be performed on a VAX computer running the VMS operating system.

SECTION 5

PREPARATION FOR DELIVERY

The implementation site for the constructed software will be the Integrated Information Support System (IISS) Test Bed. The software associated with each CPCI release will be delivered on a media which is compatible with the IISS Test Bed. The release will be clearly identified and will include instructions on procedures to be followed for installation of the release. Integration with the other IISS CPCI's will be done on the IISS TEST BED on a scheduled basis.

APPENDIX A

NDDL COMMANDS

The following is a list of NDDL commands which could be input to step one of the CDM impact analysis report. For each command (and variation) the following information is given:

1. The syntax of command.

2. The general area of impact (ES, CS, IS, CS-ES, CS-IS,or none).

3. The processing of the command and the objects impacted. In processing a percent sign "%" denotes a value from the command.

4. The type of command or message for report.

A.1 ALTER ALIAS ENTITY ec_name_1 IS ec_name_2

This command does not affect a schema. This command does not impact application programs.

A.2 ALTER ALIAS ATTRIBUTE ac_name_1 IS ac_name_2

This command does not affect a schema. This command does not impact application programs.

A.3 ALTER ATTRIBUTE CLASS ac_name

A.3.1 DOMAIN domain_name

This command affects the conceptual schema. For processing, select ac_no from ATTRIBUTE_NAME using %ac_name. Next, insert ac_no into attribute use class object. Next, select ec_no from OWNED_ATTRIBUTE using ac_no. Next, select tag_no from ATTRIBUTE_USE_CLASS using ac_no and ec_no. Next, call AUC with tag_no and include_root = TRUE. The report message is: "alter attribute %ac_name domain %domain_name"

A.3.2 DROP keyword

This command does not affect a schema. This command does not impact application programs.

A.3.3 ADD keyword

This command does not affect a schema. This command does not impact application programs.

A.3.4 Ownership to entity EC NAME ...

This command affects the conceptual schema. The resulting impact is not implemented.

A.4 ALTER CATEGORY rc name OF ec_name

This command may affect CS-IS mappings. For processing, select rc_no from CATEGORY_RELATION using % ec_name and % rc_name. Insert rc_no into relation class object. The report message is : "alter category % rc_name of % ec_name".

A.4.1 TO COMPLETE/INCOMPLETE

This clause impacts neither schemas nor applications.

A.4.2 ADD CATEGORY ec name

This clause impacts neither schemas nor applications.

A.4.3 ADD KEYWORD

This clause impacts neither schemas nor applications.

A.4.4 DROP CATEGORY ec name

Select kc_no from COMPLETE_RELATION using rc_no. Call CATMEM with kc_no. The report message is "drop category ec_name".

A.4.5 DROP KEYWORD

This clause impacts neither schemas nor applications.

A.5 ALTER DATABASE db name ...

This command has the same impact as DROP DATABASE, A.31. The report message is "alter database %db_name".

A.6 ALTER DBMS dbms name

A.6.1 ALTER HOST host id

This command affects the internal schema. This command does not impact application programs.

A.6.2 DROP HOST host id

This command affects the internal schema. This command does not impact application programs. Note: no database may be defined for the dbms on the host which is to be dropped.

A.7 ALTER DOMAIN domain name

A.7.1 ADD DATA TYPE data type name ...

This command does not affect a schema. This command does not impact application programs.

A.7.2 DROP DATA TYPE data type name

This command does not affect a schema. For processing, select domain_no from DOMAIN_CLASS using %domain_name. Next, select tag_no from ATTRIBUTE_USE_CL, ATTRIBUTE_CLASS using domain_no. For each tag_no, insert tag_no into attribute use class object. The report message is: "alter domain %domain_name".

A.7.3 ALTER DATA TYPE data type name ... [TO STANDARD]

This command has the same impact as ALTER DOMAIN DROP DATA TYPE, A.7.2.

A.8 ALTER ENTITY CLASS ec name

A.8.1 ADD KEY CLASS kc name = ac name

This command affects the conceptual schema. This command does not impact application programs.

A.8.2 ADD [OWNED] ATTRIBUTE CLASS ac name

This command affects the conceptual schema. This command does not impact application programs.

A.8.3 ADD KEYWORD keyword

This command does not affect a schema. This command does not impact application programs.

A.8.4 DROP KEY CLASS kc name

This command affects the conceptual schema. For processing select ec_no from ENTITY_NAME using %ec_name. Next select kc_no from KEYCLASS using %ec_no and %kc_name. Next call KC with kc_no to find affected attribute use class entities. The report message is: "alter entity %ec_name drop key %kc_name".

A.8.5 DROP [OWNED] ATTRIBUTE CLASS ac_name

This command affects the conceptual schema. For processing, select ec_no from ENTITY_NAME using % ec_no. Next, select ac_no, tag_no from OWNED_ATTRIBUTE using ec_no and %ac_name. Lastly, call AUC with tag_no and include_root = TRUE to find affected attribute use class entities. The report message is: "alter entitv %ec_name drop attribute %ac_name".

A.8.6 DROP KEYWORD keyword

This command does not affect a schema. This command does not impact application programs.

A.8.7 ALTER ATTRIBUTE CLASS old_tag_name TO new_tag_name

The command affects the conceptual schema. The impact is not implemented.

A.8.8 ALTER KEY kc_name [TO new_kc_name] add attribute

This command affects the conceptual schema. The impact is not implemented.

A.8.9 ALTER KEY KC_NAME [TO new_kc_name] drop attribute

This command affects the conceptual schema. The impact is not implemented.

A.8.10 ALTER KEY

ADD ATTRIBUTE ...
DROP ATTRIBUTE ...
SUBSTITUTING ...

This command affects the conceptual schema. The impact is not implemented.

A.9 ALTER FIELD field_name OF TABLE rec_name OF DATABASE db_name...

This command has the same impact as DROP FIELD, A.67. The report message is "alter field %db_name.%rec_name.%field_name".

A.10 ALTER HOST host_id

A.10.1 ADD DBMS dbms_name

This command affects the internal schema. This command does not impact application programs.

A.10.2 DROP DBMS dbms name

This command affects the internal schema. This command does not impact application programs. Note: no databases may be defined or this dbms on this host.

A.11 ALTER MAP

A.11.1 ALTER MAP ec name.tag name FOR PREFERENCE pref no 1

A.11.1.1 TO PREFERENCE pref no 2

This command affects the CS-IS mapping. For processing, select ec_no from ENTITY_NAME using %ec_name. Next, select tag_no from ATTRIBUTE_USE_CLASS using ec_no and %tag_name. Lastly, insert tag_no into attribute use class object. The report message is: "alter map %ec_name.%tag_name for preference %pref_no_1".

A.11.1.2 ADD FIELD db name.rt name.df name ...

This command affects the CS-IS mapping. This command does not impact application programs.

A.11.1.3 ADD SET db name.set name VALUE "string"

This command affects the CS-IS mapping. This command does not impact application programs.

A.11.1.4 ALTER SET db name.set name VALUE "string"

This command has the same impact as ALTER MAP, A.11.1.1.

A.11.1.5 DROP FIELD db name.rt name.df name

This command has the same impact as ALTER MAP, A.11.1.1.

A.11.1.6 DROP SET db name.set name ...

This command has the same impact as ALTER MAP, A.11.1.1.

A.11.2 ALTER MAP ec_name_1 rc_name ec_name_2

A.11.2.1 ADD SET db_name.set_name

This command affects the CS-IS mapping. This command does not impact application programs.

A.11.2.2 DROP SET db_name.set_name

This command affects the CS-IS mapping. For processing, select rc_no from RELATION_CLASS using %ec_name_1, %rc_name and %ec_name_2. Next, insert rc_no into relation class object. The report message is: "alter map %ec_name_1 %rc_name %ec_name_2".

A.11.3 ALTER MAP ec_name

A.11.3.1 ADD RECORD db_name.rec_name

This command affects the CS-IS mapping. This command does not impact application programs.

A.11.3.2 DROP RECORD db_name.rec_name

This command affects the CS-IS mapping. The impact has not been implemented.

A.11.3.3 ALLOW/DISALLOW RETRIEVAL/UPDATE

This command does not affect a schema. This command does not impact application programs.

A.12 ALTER MODEL model_name

This command does not affect a schema. This command does not impact application programs.

A.13 ALTER MODULE mod_name IN language

This command does not affect a schema. This command does not impact application programs.

A.13.1 ADD PARAMETERS AFTER|BEFORE parm_id

This command does not affect a schema. This command does not impact application programs.

A.13.2 DROP PARAMETERS PARM_ID

This command does not affect a schema. This command does not impact application programs.

A.14 ALTER PSB psb_name TO HOST host_id

This command affects the internal schema. For processing, select db_name, db_id from PSB_PCB, DATA_BASE using %psb_name. The report message is: "alter psb %psb_name to %host_id".

A.15 ALTER PARTITION integer OF ENTITY ec_name

A.15.1 ADD RECORD db_name.rec_name

This command affects the conceptual to internal schema mapping. For processing, select EN.ec_no from EC_NAME using %ec_name. Next, call ECAUC using ec_no. The report message is: "alter partition %integer of entity %ec_name add record %db_name.%rec_name".

A.15.2 DROP RECORD db_name.rec_name

This command has the same impact as A.15.1. The report message is: "alter partition %integer of entity %ec_name drop record %db_name.%rec_name".

A.16 ALTER RECORD rec_name OF DATABASE db_name

This command has the same impact as DROP RECORD, A.75. The report message is: "alter record %db_name.%rec_name".

A.17 ALTER RELATION CLASS cardinality_1 ec_name_1 rc_name cardinality_2 ec_name_2

This command has the same impact as command ALTER MAP relation, A.11.2. The report message is: "alter RC %ec_name_1 %rc_name %ec_name_2".

A.17.1 ADD MIGRATES kc_name SET tag_name_1 = tag_name_2

This command affects the conceptual schema. This command does not impact application programs.

A.17.2 ADD KEYWORD keyword

This command does not affect a schema. This command does not impact application programs.

A.17.3 DROP MIGRATES kc name

This command has the same impact as ALTER RELATION, A.17, and has additional processing, select kc_no from KEY_CLASS using %ec_name_1 and %kc_name. Next, call RC using rc_no and kc_no. The report message is: "alter RC %ec_name_1 %rc_name %ec_name_2".

A.17.4 DROP KEYWORD keyword

This command does not affect a schema. This command does not impact application programs.

A.18 ALTER UNION OF RECORD db name.rec name

A.18.1 ADD ENTITY ec name

This command impacts the conceptual to internal mapping. For processing, select ec_no from ENTITY_CLASS using %ec_name. Next, call ECAUC with ec_no. The report message is: "alter union %db_name.%rec_name add entity %ec_name".

A.18.2 DROP ENTITY ec name

This command has the same impact as the previous command, A.18.1. The report message is: "alter union %db_name.%rec_name drop entity %ec_name".

A.19 CHECK MODEL model name

This command does not affect a schema. This command does not impact application programs.

A.20 COMBINE ENTITY ec name 1 FROM MODEL model name INTO ec name 2 ON FILE 'string'

If this is an intra model combine then this command has the same impact as DROP ENTITY ec_name_1, A.66.

A.21 COMMIT

This command does not affect a schema. This command does not impact application programs.

A.22 COMPARE MODEL model name 1 WITH model name 2

This command does not affect a schema. This command does not impact application programs.

A.23 COPY ATTRIBUTE attr name 1 FROM model name TO attr name 2 ON FILE 'string'

This command affects the conceptual schema. This command does not impact application programs.

A.24 COPY DATBASE|pcb db name|ALL

This command does not affect a schema. This command does not impact application programs.

A.25 COPY DBMS dbms name|ALL INCLUDE DATABASE

This command does not affect a schema. This command does not impact application programs.

A.26 COPY DESCRIPTION desc type|ALL OF OBJECT TYPE object name [TO object name] [FROM Model from model] [DIRECTLY]

This command does not affect a schema. This command does not impact application programs.

A.27 COPY description type OF object type object id 1
FROM MODEL model name TO object id 2

This command does not affect a schema. This command does not impact application programs.

A.28 COPY ENTITY CLASS ec name 1 FROM MODEL model name TO ec name 2

This command affects the conceptual schema. This command does not impact application programs.

A.29 COPY HOST host name|ALL EXCEPT DESCRIPTION psb

This command does not affect a schema. This command does not impact application programs.

A.30 COPY MAP

This command does not affect a schema. This command does not impact application programs.

A.31 COPY MODEL

This command does not affect a schema. This command does not impact application programs.

A.32 COPY MODULE mod_name|ALL

This command does not affect a schema. This command does not impact application programs.

A.33 COPY RECORD

This command does not affect a schema. This command does not impact application programs.

A.34 COPY SET

This command does not affect a schema. This command does not impact application programs.

A.35 COPY VIEW

This command does not affect a schema. This command does not impact application programs.

A.36 CREATE ALIAS

This command does not affect a schema. This command does not impact application programs.

A.37 CREATE ATTRIBUTE

This command affects the conceptual schema. This command does not impact application programs.

A.38 CREATE CATEGORY

This command affects the conceptual schema. This command does not impact application programs.

A.39 CREATE DESCRIPTION TYPE

This command does not affect a schema. This command does not impact application programs.

A.40 CREATE DOMAIN

This command does not affect a schema. This command does not impact application programs.

A.41 CREATE ENTITY

This command affects the conceptual schema. This command does not impact application programs.

A.42 CREATE MAP

This command affects the CS-IS mapping. This command does not impact application programs.

A.43 CREATE MODEL

This command does not affect a schema. This command does not impact application programs.

A.44 CREATE PARTITION

This command has the same impact as ALTER PARTITION, A.15. The report message is: "create partition %part_name of entity %ec_name".

A.45 CREATE RELATION

This command affects the conceptual schema. This command does not impact application programs.

A.46 CREATE UNION

This command has the same impact as ALTER UNION, A.18. The report message is: "create union of record %db_name.%rec_name to entity %ec_name".

A.47 CREATE VIEW

This command affects the CS-ES mapping and external schema. This command does not impact application programs.

A.48 DEFINE ALGORITHM

This command does not impact application programs.

A.49 DEFINE DATABASE

This command affects the internal schema. This command does not impact application programs.

A.50 DEFINE DBMS

This command affects the internal schema. This command does not impact application programs.

A.51 DEFINE DATABASE

This command affects the internal schema. This command does not impact application programs.

A.52 DEFINE HOST

This command affects the internal schema. This command does not impact application programs.

A.53 DEFINE MODULE

This command does not affect application programs.

A.54 DEFINE PSB

This command affects the internal schema. This command does not impact application programs.

A.55 DEFINE RECORD

This command affects the internal schema. This command does not impact application programs.

A.56 DEFINE SET

This command affects the internal schema. This command does not impact application programs.

A.57 DESCRIBE

This command does not affect a schema. This command does not impact application programs.

A.58 DROP ALGORITHM mod_name mod_instance FOR mod_type

This command affects the conceptual to internal schema mapping. To determine the impacts one must find each of the four types of parameters: record types, data fields, data items, and attribute use classes.

For processing record types, select RT.rt_no from DATA_BASE DB, RECORD_TYPE RT, RT_PARM RP, MODULE_PARAMETER MP using %mod_name, %mod_instance and %mod_type. For each rt_no, insert rt_no into record type object.

For processing data fields, select DF.df_no from DATA_BASE DB, DATA_FIELD DF, DF_PARM DP, MODULE_PARAMETER MP using %mod_name, %mod_instance and %mod_type. For each df_no, insert df_no into data field object.

For processing data items, select DI.di_no from USER_VIEW UV, DATA_ITEM DI, DI_PARM DP, MODULE_PARAMETER MP using %mod_name, %mod_instance and %mod_type. For each di_no, insert di_no into data item object.

For processing attribute use classes, select AUC.tag_no from ATTRIBUTE_USE_CL AUC, AUC_PARM AP, MODULE_PARAMETER MP using %mod_name, %mod_instance and %mod_tye.

The report message is "drop algorithm %mod_name %mod_instance for %mod_type".

A.59 DROP ALIAS

This command does not affect a schema. This command does not impact application programs.

A.60 DROP ATTRIBUTE ac name

This command affects the conceptual schema. For processing, select ac_no from ATTRIBUTE_NAME using %ac_name. Next, select ec_no from OWN_ATTRIBUTE using ac_no. Next, select tag_no from ATTRIBUTE_USE_CLASS using ec_no and ac_no. Lastly, call AUC with tag_no and include_root = TRUE to determine affected attribute use class entities. The report message is: "drop attribute %ac_name".

A.61 DROP CATEGORY ec name rc name

This command has the same impact as the ALTER MAP EC_NAMES RC_NAME command A.11.2. For additional processing, select kc_no from COMPLETE_RELATION using rc_no and call RC with rc_no and kc_no. The report message is: "drop category % ec_name % rc_name.

A.62 DROP db type DATABASE NAMED db name

This command affects the internal schema. For processing, select db_id from DATABASE using %db_name and %db_type. Next, insert the db_id into the database object. The report message is: "drop database %db_name".

A.63 DROP DBMS

This command affects the internal schema. This command does not affect application programs. Note: a dbms cannot be cross referenced with a host or a database when dropped.

A.64 DROP DESCRIPTION TYPE

This command does not affect a schema. This command does not affect application programs.

A.65 DROP DOMAIN domain name

This command does not affect a schema. This command does not impact application programs. Note: dropping a domain is allowed only if nothing references it.

A.66 DROP ENTITY ec name

This command affects the conceptual schema. For processing, select ec_no from ENTITY_NAME using %ec_name. Next, call EC with ec_no. The report message is: "drop entity %ec_name".

A.67 DROP FIELD df id ... OF RECORD rt id OF DATABASE db name

This command affects the internal schema. For processing, select db_id from DATABASE using %db_id. Next, select df_no from DATA_FIELD using db_id, rt_id, and %df_name. Lastly, insert df_no into the data field object. The report message is: "drop field %db_name.%rt_id.%df_name".

A.68 DROP HOST

This command affects the internal schema. This command does not affect application programs. Note: a host cannot be cross referenced with a dbms or a database when dropped.

A.69 DROP KEYWORD keyword

This command does not affect a schema. This command does not impact application programs.

A.70 DROP MAP

A.70.1 DROP MAP ec name.tag name FOR PREFERENCE pref no

This command has the same impact as ALTER MAP ec_name.tag_name, A.11.1.1.

A.70.2 DROP MAP ec name 1 rc name ec name 2

This command has the same impact as ALTER MAP ec_name_1 rc_name ec_name_2 A.11.2.2.

A.71 DROP MODEL model_name

This command affects the conceptual schema. This command does not impact application programs. Note: One may not drop the integrated model.

A.72 DROP MODULE

This command does not affect a schema. This command does not affect application programs. Note: a module may not be associated with a complex mapping when dropped.

A.73 DROP PARTITION integer OF ENTITY ec_name

This command has the same impact as ALTER PARTITION, A.15. The report message is "drop partition %integer of entity %ec_name".

A.74 DROP PSB psb_name

This command affects the internal schema. This command does not affect application programs. Note: a PSB's database must be dropped prior to dropping a PSB.

A.75 DROP RECORD rt_id OF DATABASE db_name

This command affects the internal schema. For processing, select db_id from DATABASE using %db_name. Next, select rt_no from RECORD_TYPE using %rt_id. Lastly, insert rt_no into the record type object. The report message is: "drop record %db_name.%rt_id".

A.76 DROP RELATION ec_name_1 rc_name ec_name_2

This command has the same impact as command ALTER MAP EC_NAME_1 RC_NAME EC_NAME_2, A.11.2 and for additional processing, select kc_no from COMPLETE_RELATION using rc_no. Next, call RC with rc_no and kc_no. The report message is: "drop relation %ec_name_1 %rc_name %ec_name_2".

A.77 DROP SET set_id OF DATABASE db_name

This command affects the internal schema. For processing, select db_id from DATABASE using %db_name. Next, select set_no from RECORD_SET using db_id and %set_no. Lastly, insert set_no into the record set object. The report message is: "drop set %db_name.%set_id".

A.78 DROP UNION OF RECORD db_name.rec_name

This command has the same impact as DROP RECORD, A.75. The report message is: "drop union of record %db_name.%rec_name".

A.79 DROP VIEW view_name

This command affects the CS-ES mapping and the external schema. For processing, select view_no from USER_VIEW using %view_name. Next, insert view_no into user view object. The report message is: "drop view %view_name".

A.80 HALT

This command does not affect a schema. This command does not impact application programs.

A.81 MERGE MODEL model_name_1 WITH model_name_2 INTO model_name_3ON FILE 'string'

This command does not affect a schema. This command does not impact application programs. The action is to write NDDL to a file without any internal updates.

A.82 RENAME ENTITY

This command does not affect a schema. This command does not impact application programs.

A.83 RENAME ATTRIBUTE

This command does not affect a schema. This command does not impact application programs.

A.84 RENAME KEYWORD

This command does not affect a schema. This command does not impact application programs.

A.85 RENAME MODEL

This command does not affect a schema. This command does not impact application programs.

A.86 RENAME DOMAIN

This command does not affect a schema. This command does not impact application programs.

A.87 RENAME VIEW

This command has the same impact as command DROP VIEW, A.79. The report message is: "rename %view_name_1 to %view_name_2".

A.88 RENAME RELATION

This command affects the conceptual schema. This command does not impact application programs.

A.89 RENAME DATAITEM view name.di name 1 TO di name 2

This command affects the external schema. For processing, select DI.di_no from DATA_ITEM DI, USER_VIEW UV where %view_name %di_name_1. Next, insert di_no into data item object.. The report message is: "rename data item view_name.di_name_1 to %di_name_2".

A.90 RENAME HOST host name 1 TO host name 2

This command affects the internal schema. For processing, select DB.db_id from DATABASE DB using %host_name_1. For each db_id, insert db_id into database object. The report message is: "rename host %host_name_1 to %host_name_2".

A.91 RENAME DATABASE db name 1 TO db name 2

This command affects the internal schema. For processing, select DB.db_id from DATABASE DB using %db_name_1. Next, insert db_id into database object. The report message is: "rename database %db_name_1 to %db_name_2".

A.92 RENAME FIELD db name.rec name.df name 1 to df name 2

This command affects the internal schema. For processing, select DF.df_no from DATABASE DB, RECORD_TYPE RT, DATA_FIELD DF using %db_name, %rec_name and %df_name_1. The report message is: "rename datafield %db_name.%rcc_name.%df_name_1 and %df_name_2".

A.93 RENAME DATATYPE

This command does not affect a schema. This command does not affect application programs.

A.94 RENAME SET db name.set name 1 TO set name 2

This command affects the internal schema. For processing, select RS.set_no from DATA_BASE DB, RECORD_SET RS using %db_name and %set_name_1. Next, insert set_no into record set object. The report message is "rename set %db_name.%set_name_1 to %set_name2".

A.95 RENAME RECORD db name.rec name 1 TO rec name 2

This command effects the internal schema. For processing, select RT.rt_no from DATA_BASE DB, RECORD_TYPE RT using %db_name and %rec_name1. Next, insert re_no into recorddd type object. The report message is: "rename record %db_name.%rec_name_1 to %rec_name_2".

☆U.S. GOVERNMENT PRINTING OFFICE: 1992 - 648-127/62342